

# Derivative Computations and Robust Standard Errors for Linear Mixed Effects Models in **lme4**

Ting Wang

American Board of Anesthesiology

Edgar C. Merkle

University of Missouri

---

## Abstract

While robust standard errors and related facilities are available in R for many types of statistical models, the facilities are notably lacking for models estimated via **lme4**. This is because the necessary statistical output, including the Hessian and casewise gradient of random effect parameters, is not immediately available from **lme4** and is not trivial to obtain. In this article, we supply and describe two new functions to obtain this output from Gaussian mixed models: `estfun.lmerMod()` and `vcov.full.lmerMod()`. We discuss the theoretical results implemented in the code, focusing on calculation of robust standard errors via package **sandwich**. We also use the **Sleepstudy** data to illustrate the code and compare it to a benchmark from package **lavaan**.

*Keywords:* Linear mixed model, scores, Huber-White sandwich estimator, robust standard error, **lme4**.

---

## 1. Introduction

Package **lme4** (Bates, Mächler, Bolker, and Walker 2015) is widely used to estimate a variety of generalized linear mixed models. Despite its popularity, the package does not provide certain results related to estimated random effect parameters, which makes it difficult to obtain robust standard errors and other statistical tests. This absence is at least partially related to the fact that **lme4** does not directly estimate models via likelihood maximization, but rather employs a penalized least squares approach that leads to ML (or REML) estimates (Bates *et al.* 2015). While this approach eases model estimation, it also makes it more difficult to obtain derivatives (first and second) of the likelihood from a fitted model (which are required for, e.g., the Huber-White sandwich estimator). While it is possible to instead utilize the robust estimation methods from package **robustlmm** (Koller 2016), we are interested in directly obtaining the full, robust variance covariance matrix (as opposed to only the diagonal entries) from models estimated by **lme4**. Thus, the goal of this paper is to describe functions that compute these derivatives for objects of class `lmerMod`: `estfun.lmerMod()` and `vcov.full.lmerMod()`. We also briefly discuss derivatives associated with models of class `glmerMod`, though we do not currently have code for these models (the computations are significantly more difficult due to the lack of analytic results).

To aid in discussion of the derivatives, we consider the computation of the Huber-White sandwich estimator (Eicker 1967; White 1980; Huber 1967). We specifically build functions that can be sent to package **sandwich** (Zeileis 2004, 2006), letting that package do the Huber-White computations. The paper proceeds as follows. We first introduce background notation along

with the general Huber-White sandwich estimator approach. Next, we derive expressions for linear mixed models' casewise and clusterwise first derivatives and Hessians, including the estimated random effect parameters. Next, we illustrate our implementation via the sleep study data (Belenky, Wesensten, Thorne, Thomas, Sing, Redmond, Russo, and Balkin 2003) included with **lme4**, comparing our results to a benchmark from **lavaan** (Rosseel 2012). Finally, we discuss use and extension of our functions beyond robust standard errors.

## 2. Background

In this section, we provide background notation and detail on the linear mixed model and on the Huber-White sandwich estimator.

### 2.1. Linear Mixed Model

Let  $\mathbf{y}$  include  $n$  observations clustered in  $J$  groups. The linear mixed model can be written as

$$\mathbf{y}|\mathbf{b} \sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b}, \mathbf{R}) \quad (1)$$

$$\mathbf{b} \sim N(\mathbf{0}, \mathbf{G}) \quad (2)$$

$$\mathbf{R} = \sigma_r^2 \mathbf{I}_n, \quad (3)$$

where  $\mathbf{y}$  is the observed data vector of length  $n$ ;  $\mathbf{X}$  is an  $n \times p$  matrix of fixed covariates;  $\boldsymbol{\beta}$  is the fixed effect vector of length  $p$ ;  $\mathbf{Z}$  is an  $n \times q$  design matrix of random effects; and  $\mathbf{b}$  is the random effect vector of length  $q$  (where  $q$  is typically some multiple of  $J$ ).

The vector  $\mathbf{b}$  is assumed to follow a normal distribution with mean  $\mathbf{0}$  and covariance matrix  $\mathbf{G}$ , where  $\mathbf{G}$  is a block diagonal matrix composed of random effect parameters and their covariances. The residual covariance matrix,  $\mathbf{R}$ , is the product of the residual variance  $\sigma_r^2$  and an identity matrix of dimension  $n$ . We further define  $\boldsymbol{\sigma}^2$  to be a vector of length  $K$ , containing all variance/covariance parameters (including those of the random effects and the residual). Thus, the matrix  $\mathbf{G}$  has  $(K - 1)$  unique elements. For example, in a model with two random effects that are allowed to covary,  $\boldsymbol{\sigma}^2$  is a vector of length 4 (i.e.,  $K = 4$ ). The first three elements correspond to the unique entries of  $\mathbf{G}$ , which are commonly expressed as  $\sigma_0^2$ ,  $\sigma_0\sigma_1$ , and  $\sigma_1^2$ . The last component is then the residual variance  $\sigma_r^2$ .

Based on Equations (1), (2) and (3), the marginal distribution of the LMM is

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{V}), \quad (4)$$

where

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \sigma_r^2 \mathbf{I}_n. \quad (5)$$

Therefore, the marginal likelihood can be expressed as

$$\ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y}) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log(|\mathbf{V}|) - \frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (6)$$

## 2.2. Huber-White Sandwich Estimator

Let  $\mathbf{y}_{c_j}$  contain the observations within cluster  $c_j$ . As shown in the previous section, each entry of  $\mathbf{y}_{c_j}$  is conditionally independent of the others given the random effects  $\mathbf{b}$ . Observations in different groups are unconditionally independent. This allows us to write

$$\ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y}) = \sum_{j=1}^J \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y}_{c_j}) \quad (7)$$

where  $\ell()$  is defined in (6).

The first and second partial derivatives of  $\ell$  w.r.t.  $\boldsymbol{\xi} = (\boldsymbol{\sigma}^2 \ \boldsymbol{\beta})^\top$  are therefore

$$\ell'(\boldsymbol{\xi}; \mathbf{y}) = \sum_{j=1}^J \frac{\partial \ell(\boldsymbol{\xi}; \mathbf{y}_{c_j})}{\partial \boldsymbol{\xi}} = \sum_{j=1}^J \sum_{i \in c_j} s_i(\boldsymbol{\xi} | y_i) \quad (8)$$

$$\ell''(\boldsymbol{\xi}; \mathbf{y}) = \sum_{j=1}^J \frac{\partial^2 \ell(\boldsymbol{\xi}; \mathbf{y}_{c_j})}{\partial \boldsymbol{\xi}^2}, \quad (9)$$

where the  $s_i()$  are often called *scores* and have been studied in other contexts (e.g., Wang, Merkle, and Zeileis 2014; Zeileis and Hornik 2007).

Inference about  $\boldsymbol{\xi}$  relies on a central limit theorem:

$$\sqrt{n}(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}) \xrightarrow{d} N(\mathbf{0}, \mathbf{V}(\boldsymbol{\xi})), \quad (10)$$

where  $\xrightarrow{d}$  denotes convergence in distribution. The traditional estimate of  $\mathbf{V}(\boldsymbol{\xi})$  relies on Equation (9), whereas the Huber-White sandwich estimator of  $\mathbf{V}(\boldsymbol{\xi})$  is defined as (e.g., Freedman 2006; White 1980; Zeileis 2006):

$$\mathbf{V}(\hat{\boldsymbol{\xi}}) = (-\mathbf{A})^{-1} \mathbf{B} (-\mathbf{A})^{-1}, \quad (11)$$

where  $\mathbf{A} = \ell''(\hat{\boldsymbol{\xi}}; \mathbf{y})$  and  $\mathbf{B} = \text{Cov}(\ell'(\hat{\boldsymbol{\xi}}; \mathbf{y}))$ . The square roots of the diagonal elements of  $\mathbf{V}$  are the so-called “robust standard errors”.

When the model is correctly specified, the Huber-White sandwich estimator corresponds to the Fisher information matrix. However, the estimator is often used in non-*i.i.d.* samples to “correct” the information matrix for misspecification (e.g., Freedman 2006). While mixed models explicitly handle lack of independence via random effects, the Huber-White estimators can still be applied to these models to address remaining model misspecifications.

To construct the Huber-White sandwich estimator,  $\mathbf{A}$  can be directly obtained from Equation (9), whereas  $\mathbf{B}$  can be constructed via (e.g., Freedman 2006):

$$\mathbf{B} = \sum_{j=1}^J \left[ \sum_{i \in c_j} s_i(y_i | \boldsymbol{\xi}) \right]^\top \left[ \sum_{i \in c_j} s_i(y_i | \boldsymbol{\xi}) \right]. \quad (12)$$

Thus, our goal here is to obtain the “score” terms  $s_i(\boldsymbol{\xi}; y_i)$  ( $i = 1, \dots, n$ ) and the Hessian

$\ell''(\boldsymbol{\xi}; \mathbf{y})$  using the marginal likelihood from (6).

### 3. Derivative Computations for the Linear Mixed Model

In this section, we first discuss analytic results involving the linear mixed model's first and second derivatives. We then illustrate how these derivatives can be obtained from an object of class `lmerMod`.

#### 3.1. Scores

Based on the objective function from Equation (6), we derive the score function  $s_i()$  for each observation w.r.t. the parameter vector  $\boldsymbol{\xi} = (\boldsymbol{\sigma}^2, \boldsymbol{\beta})^\top$ . We focus separately on  $\boldsymbol{\sigma}^2$  and on  $\boldsymbol{\beta}$  below.

##### *Scores for $\boldsymbol{\sigma}^2$*

The gradient with respect to the  $k^{\text{th}}$  entry of  $\boldsymbol{\sigma}^2$  ( $k = 1, 2, 3, \dots, K$ ) is (Stroup 2012, p. 136–137):

$$\frac{\partial \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \sigma_k^2} = -\frac{1}{2} \text{tr} \left[ \mathbf{V}^{-1} \frac{\partial \mathbf{V}}{\partial \sigma_k^2} \right] + \frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{V}^{-1} \left( \frac{\partial \mathbf{V}}{\partial \sigma_k^2} \right) \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \quad (13)$$

where  $\mathbf{V}$  is defined in (5). This gradient sums over  $i$ , whereas the scores are defined for each observation  $i$ . Based on matrix theory, the scores  $s_i(\boldsymbol{\sigma}^2; y_i)$  can be obtained by taking the diagonal elements of the first term of Equation (13) and the Kronecker product of the second term. This allows us to write the scores w.r.t.  $\sigma_k^2$  as:

$$s(\sigma_k^2; \mathbf{y}) = -\frac{1}{2} \text{diag} \left[ \mathbf{V}^{-1} \frac{\partial \mathbf{V}}{\partial \sigma_k^2} \right] + \left\{ \frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{V}^{-1} \left( \frac{\partial \mathbf{V}}{\partial \sigma_k^2} \right) \mathbf{V}^{-1} \right\} \otimes (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (14)$$

##### *Scores for $\boldsymbol{\beta}$*

For the fixed effect parameter  $\boldsymbol{\beta}$ , the gradient is:

$$\frac{\partial \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (15)$$

Based on matrix theory,  $s(\boldsymbol{\beta}; \mathbf{y})$  can be obtained by taking the the Kronecker product of these terms:

$$s(\boldsymbol{\beta}; \mathbf{y}) = \left\{ \mathbf{X}^\top \mathbf{V}^{-1} \right\} \otimes (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (16)$$

Thus, the score function required by Equation (11) can be expressed as a matrix whose columns consist of the results from Equations (14) and (16).

These equations provide scores for each observation  $i$ , and we can construct the clusterwise scores by summing scores within each cluster. In situations with one grouping (clustering)

variable, the clusterwise scores can be obtained from our `estfun.lmerMod()` function via the default argument `level=2`. In situations with multiple grouping variables (i.e., crossed random effects, three-level models), the function returns an error (more detail in the Discussion section). The casewise scores, on the other hand, can be retrieved for all models via the argument `level = 1`.

### 3.2. Hessian

The Hessian, which is the  $\mathbf{A}$  matrix in Equation (11), can often be obtained in R with the help of the `vcov()` function. However, package **lme4** does not provide a Hessian that includes both the fixed and random effect parameters. Thus, the derivation of this matrix requires special attention.

To obtain this Hessian, we can divide the matrix  $\mathbf{A}$  into the following four blocks:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial^2 \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} & \frac{\partial^2 \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\sigma}^2} \\ \frac{\partial^2 \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\sigma}^2 \partial \boldsymbol{\beta}} & \frac{\partial^2 \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\sigma}^2 \partial \boldsymbol{\sigma}^2} \end{bmatrix},$$

where  $\boldsymbol{\beta}$  contains all fixed parameters and  $\boldsymbol{\sigma}^2$  contains all variance-covariance parameters (in variance-covariance scale) in the linear mixed model. To facilitate the analytical derivations, we index the above four blocks as

$$\mathbf{A} = \begin{bmatrix} \text{Block 1} & \text{Block 3} \\ \text{Block 2} & \text{Block 4} \end{bmatrix}.$$

Block 1 is straightforward, which is provided by `-solve(vcov())` in **lme4**. Block 4 is obtained via (Stroup 2012, Equation (4.26))

$$\frac{\partial^2 \ell(\boldsymbol{\sigma}, \mathbf{y}, \boldsymbol{\beta})}{\partial \sigma_{k_1}^2 \partial \sigma_{k_2}^2} = \left( \frac{1}{2} \right) \text{tr} \left[ \mathbf{V}^{-1} \left( \frac{\partial \mathbf{V}}{\partial \sigma_{k_1}^2} \right) \mathbf{V}^{-1} \left( \frac{\partial \mathbf{V}}{\partial \sigma_{k_2}^2} \right) \right], \quad (17)$$

where  $k_1 \in 1, \dots, K$  and  $k_2 \in 1, \dots, K$ .

Finally, Block 2 (which is the transpose of Block 3) can be obtained by focusing on Equation (4.24) of Stroup (2012) and taking derivatives w.r.t.  $\boldsymbol{\beta}$ . Using the identity from Equation (84) of Petersen and Pedersen (2012), this allows us to derive Block 3 as

$$\frac{\partial^2 \ell(\boldsymbol{\sigma}^2, \boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\sigma}^2 \partial \boldsymbol{\beta}} = -\mathbf{X}^\top \mathbf{V}^{-1} \left( \frac{\partial \mathbf{V}}{\partial \boldsymbol{\sigma}^2} \right) \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (18)$$

Thus, we have expressed the necessary derivatives as functions of model matrices and derivatives of the marginal variance  $\mathbf{V}$ . We now briefly describe how these derivatives and model matrices can be obtained from an object of class `lmerMod`.

## 4. Application to `lmerMod` Objects

In this section, we describe how to obtain all the quantities needed to compute the scores and Hessian from an `lmerMod` object. The data and model matrices  $\mathbf{y}$ ,  $\mathbf{X}$ ,  $\boldsymbol{\beta}$ , and  $\mathbf{Z}$  can be obtained directly from **lme4** via `getME()`. The only remaining components, then, are  $\mathbf{V}$  and  $\partial\mathbf{V}/\partial\sigma^2$ . In the following, we focus on how to indirectly obtain these components.

In the **lme4** framework, the random effects covariance matrix  $\mathbf{G}$  is modeled via (Bates *et al.* 2015, Equation 4)

$$\mathbf{G} = \boldsymbol{\Lambda}_\theta \boldsymbol{\Lambda}_\theta^\top \sigma_r^2, \quad (19)$$

where  $\boldsymbol{\Lambda}_\theta$  is a  $q \times q$  lower diagonal matrix, called the *relative covariance factor*. It can be seen as a Cholesky decomposition of  $\mathbf{G}/\sigma_r^2$ . The dimension of  $\boldsymbol{\Lambda}_\theta$  is the same as that of  $\mathbf{G}$ . Additionally, the position of  $\sigma_k^2$  in  $\mathbf{G}$  is the same as the position of  $\theta_k$  in  $\boldsymbol{\Lambda}_\theta$ .

Inserting Equation (19) into Equation (5), we can express  $\mathbf{V}$  as

$$\mathbf{V} = (\mathbf{Z} \boldsymbol{\Lambda}_\theta \boldsymbol{\Lambda}_\theta^\top \mathbf{Z}^\top + \mathbf{I}_n) \sigma_r^2. \quad (20)$$

Equation (20) is mathematically equivalent to Equation (5), but it has the computational advantage of incorporating complicated models (i.e., models with crossed random effects).

Using Equation (5), the term  $\partial\mathbf{V}/\partial\sigma_k^2$  can usually be expressed as

$$\mathbf{Z} \frac{\partial\mathbf{G}}{\partial\sigma_k^2} \mathbf{Z}^\top, \quad (21)$$

so long as  $\sigma_k^2$  is not the residual variance. The partial derivative  $\frac{\partial\mathbf{G}}{\partial\sigma_k^2}$  is then a matrix of the same dimension as  $\mathbf{G}$ , with an entry of 1 corresponding to the location of  $\sigma_k^2$  and 0 elsewhere.

Because the location of  $\sigma_k^2$  within  $\mathbf{G}$  matches its location within  $\boldsymbol{\Lambda}_\theta$ , we can use  $\boldsymbol{\Lambda}_\theta$  to facilitate computation of  $\partial\mathbf{V}/\partial\sigma_k^2$ . The only trick is that  $\mathbf{G}$  is symmetric, whereas  $\boldsymbol{\Lambda}_\theta$  is lower diagonal.

The code below illustrates implementation of this strategy, where `object` is a fitted model of class `lmerMod`. We use `forceSymmetric()` to convert the lower diagonal information from  $\boldsymbol{\Lambda}_\theta$  into the symmetric  $\mathbf{G}$ .

```
> ## "object" is a fitted model of class lmerMod.
> parts <- getME(object, "ALL")
> uluti <- length(parts$theta)
> devLambda <- vector("list", uluti)
> devV <- vector("list", (uluti+1))
>
> for (k in 1:uluti){
```

```

+   devLambda[[k]] <- parts$Lambda
+   devLambda[[k]][which(devLambda[[k]] != parts$theta[k])] <- 0
+   devLambda[[k]][which(parts$Lambda == parts$theta[k])] <- 1
+   devLambda[[k]] <- Matrix::forceSymmetric(devLambda[[k]], uplo="L")
+   devV[[k]] <- (parts$Z %*% devLambda[[k]] %*% parts$Zt)
+ }

```

Finally, for the derivative with respect to the residual variance, it is obvious that  $\partial \mathbf{V} / \partial \sigma_r^2 = \mathbf{I}_n$  so long as  $\mathbf{R} = \sigma_r^2 \mathbf{I}$  (also see [Stroup 2012](#), p. 137).

The above results are sufficient for obtaining the derivatives necessary for computing the Huber-White sandwich estimator and for carrying out additional methods (see the Discussion section). The application below focuses on the Huber-White estimator.

## 5. Application

In this section, we illustrate how our code can be used to obtain clusterwise robust standard errors for the **Sleepstudy** data ([Belenky et al. 2003](#)) included in **lme4**. This dataset includes 18 subjects participating in a sleep deprivation study, where each subject's reaction time was monitored for 10 consecutive days. The reaction times are nested by subject and continuous in measurement, hence the linear mixed model.

We first load package **lme4**, along with the functions that we developed.

```

> library("lme4")
> source("estfun.lmerMod.R")
> source("vcov.full.lmerMod.R")

```

Next, we fit a model with **Days** as the covariate, including random intercept and slope effects that are allowed to covary. There are six free model parameters: the fixed intercept and slope  $\beta_0$  and  $\beta_1$ , the random variance and covariances  $\sigma_0^2$ ,  $\sigma_1^2$ , and  $\sigma_{01}$ , and the residual variance  $\sigma_r^2$ .

```

> library("lme4")
> lme4fit <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy, REML=FALSE)

```

This particular model can also be estimated as a structural equation model via package **lavaan**, facilitating the comparison of our results with a benchmark. We first convert the data to wide format and then specify/estimate the model:

```

> testwide <- reshape2::dcast(sleepstudy, Subject ~ Days, value.var = "Reaction")
> names(testwide)[2:11] <- paste("d", 1:10, sep="")
>
> latent <- '
+   i =~ 1*d1 + 1*d2 + 1*d3 + 1*d4 + 1*d5
+         + 1*d6 + 1*d7 + 1*d8 + 1*d9 + 1*d10
+
+   s =~ 0*d1 + 1*d2 + 2*d3 + 3*d4 + 4*d5
+         + 5*d6 + 6*d7 + 7*d8 + 8*d9 + 9*d10
+

```

```

+      d1 ~~ evar*d1
+      d2 ~~ evar*d2
+      d3 ~~ evar*d3
+      d4 ~~ evar*d4
+      d5 ~~ evar*d5
+      d6 ~~ evar*d6
+      d7 ~~ evar*d7
+      d8 ~~ evar*d8
+      d9 ~~ evar*d9
+      d10 ~~ evar*d10
+
+      ## reparameterize as sd
+      sdevar := sqrt(evar)
+      i ~~ ivar*i
+      isd := sqrt(ivar)
+      '
> lavaanfit <- growth(latent, data = testwide, estimator="ML")

```

The parameter estimates from the two packages (not shown) all agree to at least three decimal places.

### *Scores*

The analytic casewise and clusterwise scores are obtained via `estfun.lmerMod()`, using the arguments `level = 1` and `level = 2`, respectively. The sum of scores (either casewise or clusterwise) equals the gradient, which is close to zero at the ML estimates.

```

> score1 <- estfun.lmerMod(lme4fit, level = 1)
> gradients1 <- colSums(score1)
> gradients1

```

	(Intercept)	Days
	2.400794e-14	2.384870e-13
cov_Subject.(Intercept)	cov_Subject.Days.(Intercept)	
	2.943434e-09	4.185322e-08
cov_Subject.Days		residual
	8.290046e-08	-7.384829e-09

```

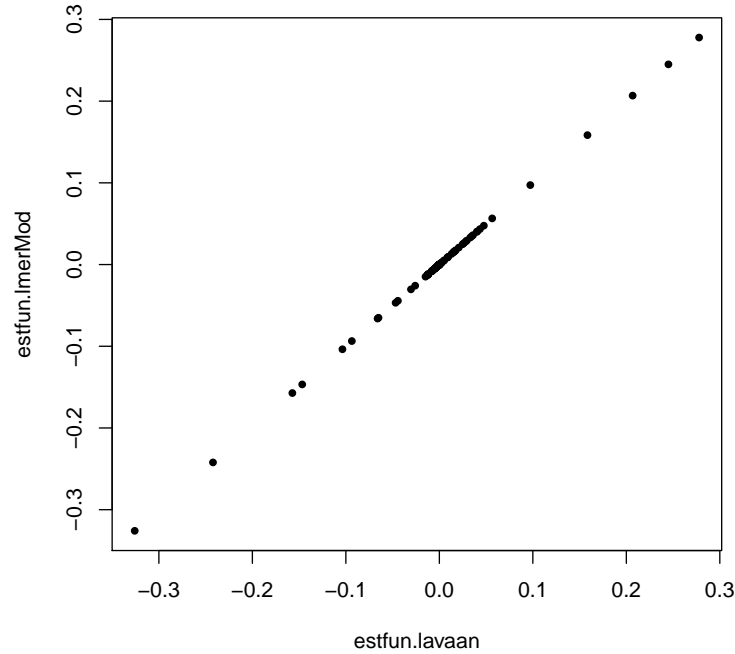
> score2 <- estfun.lmerMod(lme4fit, level= 2)
> gradients2 <- colSums(score2)
> gradients2

```

	(Intercept)	Days
	2.400326e-14	2.385054e-13
cov_Subject.(Intercept)	cov_Subject.Days.(Intercept)	
	2.943434e-09	4.185322e-08
cov_Subject.Days		residual
	8.290046e-08	-7.384829e-09



Figure 1: Comparison of scores obtained via `estfun.lavaan` and `estfun.lmerMod`. The y-axis represents analytical, clusterwise scores obtained from `estfun.lmerMod`, and the x-axis represents clusterwise scores obtained from `estfun.lavaan`.



The clusterwise scores are also provided by `estfun.lavaan()` in **lavaan**. Figure 1 presents a comparison between the clusterwise scores obtained from `estfun.lmerMod()` and `estfun.lavaan()`, showing they are nearly identical.

### *Variance Covariance Matrices*

We also compare the variance covariance matrix calculated via our **lme4** second derivatives to the `vcov` output of **lavaan**. The results are displayed in Table 1. The maximum of the absolute difference for all components in the variance covariance matrix is 0.07. This minor difference is due to the fact that we directly compute derivatives w.r.t.  $\partial\sigma^2\partial\beta$  (corresponding to Block 2 and Block 3), whereas **lavaan** forces these components to be 0 (these components are generally very small, around  $10^{-10}$ ).

Finally, the clusterwise Huber-White sandwich estimator is shown in Table 2, which is comparable to the one provided by **lavaan**. The maximum of the absolute difference for all components in the variance covariance matrix is 0.05. The minor difference is again caused by the second derivatives that are computed, versus being forced to 0.

## 6. Discussion

In this paper, we illustrated how to obtain the Huber-White sandwich estimator of estimated parameters arising from objects of class `lmerMod`. This required us to derive observational (and clusterwise) scores for fixed and random parameters (leading to the “meat”) as well as a Hessian matrix that included random effect variances and covariances (leading to the “bread”). As further described below, these functions can be used and extended to obtain various related statistical metrics.

### 6.1. Statistical Tests

The scores derived in this paper can potentially be used to carry out a variety of score-based statistical tests. For example, the “fluctuation test” framework discussed by Zeileis and Hornik (2007), Merkle and Zeileis (2013), and others generalizes the traditional score (Lagrange multiplier) test and is used to detect parameter instability across orderings of observations. The tests have been critical for the development of model-based recursive partitioning procedures available via packages such as `partykit` (Hothorn and Zeileis 2015).

The code that we present here facilitates application of score-based tests to linear mixed models, because the tests described in the previous paragraph are available via object-oriented R packages. A challenge involves the fact that much of the above theory requires observations to be independent. Thus, while we can test parameter instability across independent clusters, it is more difficult to test for instability across correlated observations within a cluster. A related issue, further described below, arises when we attempt to apply sandwich estimators to models with crossed random effects.

Finally, the work presented here also allows us to carry out Vuong tests (Vuong 1989) of non-nested linear mixed models (e.g., Merkle, You, and Preacher 2016). In particular, package `nonnest2` (Merkle and You 2016) can be extended to make use of the above results.

### 6.2. Crossed Random Effects

The “independence” challenges described in the previous section translate to the setting of models with (partially) crossed random effects. These correspond to situations where there are at least two unique variables defining clusters (for example, clusters defined by primary school attended and by secondary school attended). In this case, we cannot simply sum scores within a cluster to obtain independent, clusterwise scores. This is because observations in different clusters on the first grouping variable may be in the same cluster on the second grouping variable. Thus, it is unclear how the statistical machinery developed for independent observations (e.g., robust standard errors, instability tests) can transfer to models with partially crossed random effects. That is, while our `estfun.lmerMod()` code can return observation-level scores and the `vcov.full.lmerMod` can return the full variance covariance matrix of all `lmerMod` objects, it is unclear how to further use these results for models with crossed random effects.

### 6.3. GLMM

Finally, the procedures described here for scores, Hessians, and sandwich estimators can be extended to generalized linear mixed models estimated via `glmer()`. The technical difficulty involved with this extension is the observational scores. In the linear mixed model, we can derive the analytical scores for each observation because we know that the marginal distribution is normal. In the GLMM, the marginal distribution is typically unknown, and we require integral approximation methods (e.g., quadrature or the Laplace approximation) to obtain the scores and second derivatives. Combination of these integral approximation methods with the **lme4** penalized least squares approach presents a challenge that we have not yet overcome. We hope to do so in the future.

## 7. Acknowledgments

This research was partially supported by NSF grant 1460719. To obtain `estfun.lmerMod` and `vcov.full.lmerMod` functions, please visit <http://semtools.r-forge.r-project.org/>.

## References

- Bates D, Mächler M, Bolker B, Walker S (2015). “Fitting Linear Mixed-Effects Models Using **lme4**.” *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.
- Belenky G, Wesensten NJ, Thorne DR, Thomas ML, Sing HC, Redmond DP, Russo MB, Balkin TJ (2003). “Patterns of Performance Degradation and Restoration During Sleep Restriction and Subsequent Recovery: A Sleep Dose-Response Study.” *Journal of Sleep Research*, **12**(1), 1–12.
- Eicker F (1967). “Limit Theorems for Regressions with Unequal and Dependent Errors.” In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 59–82.
- Freedman DA (2006). “On the So-Called ‘Huber Sandwich Estimator’ and ‘Robust Standard Errors’.” *The American Statistician*, (4), 299–302.
- Hothorn T, Zeileis A (2015). “**partykit**: A Modular Toolkit for Recursive Partytioning in R.” *Journal of Machine Learning Research*, **16**, 3905–3909. URL <http://jmlr.org/papers/v16/hothorn15a.html>.
- Huber PJ (1967). “The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions.” In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 221–233.
- Koller M (2016). “**robustlmm**: An R Package for Robust Estimation of Linear Mixed-Effects Models.” *Journal of Statistical Software*, **75**(6), 1–24. doi:10.18637/jss.v075.i06.
- Merkle EC, You D (2016). *nonnest2: Tests of Non-Nested Models*. R package version 0.4-1, URL <https://cran.r-project.org/web/packages/nonnest2/index.html>.

- Merkle EC, You D, Preacher KJ (2016). “Testing Nonnested Structural Equation Models.” *Psychological Methods*, **21**(2), 151–163.
- Merkle EC, Zeileis A (2013). “Tests of Measurement Invariance without Subgroups: A Generalization of Classical Methods.” *Psychometrika*, **78**, 59–82.
- Petersen KB, Pedersen MS (2012). “The Matrix Cookbook.” Version 20121115, URL <http://www2.imm.dtu.dk/pubdb/p.php?3274>.
- Rosseel Y (2012). “**lavaan**: An R Package for Structural Equation Modeling.” *Journal of Statistical Software*, **48**(2), 1–36. URL <http://www.jstatsoft.org/v48/i02/>.
- Stroup WW (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. CRC Press.
- Vuong QH (1989). “Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses.” *Econometrica*, **57**, 307–333.
- Wang T, Merkle E, Zeileis A (2014). “Score-Based Tests of Measurement Invariance: Use in Practice.” *Frontiers in Psychology*, **5**(438), 1–11.
- White H (1980). “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity.” *Econometrica: Journal of the Econometric Society*, **48**(4), 817–838.
- Zeileis A (2004). “Econometric Computing with HC and HAC Covariance Matrix Estimators.” *Journal of Statistical Software*, **11**(10), 1–17. URL <http://www.jstatsoft.org/v11/i10/>.
- Zeileis A (2006). “Object-Oriented Computation of Sandwich Estimators.” *Journal of Statistical Software*, **16**(9), 1–16. URL <http://www.jstatsoft.org/v16/i09/>.
- Zeileis A, Hornik K (2007). “Generalized M-Fluctuation Tests for Parameter Instability.” *Statistica Neerlandica*, **61**, 488–508.

### Affiliation:

Ting Wang  
 The American Board of Anesthesiology  
 4208 Six Forks Road, Suite 1500  
 Raleigh, NC 27609-5765  
 E-mail: [ting.wang@theaba.org](mailto:ting.wang@theaba.org)

Edgar C. Merkle  
 Department of Psychological Sciences  
 University of Missouri  
 28A McAlester Hall  
 Columbia, MO, USA 65211  
 E-mail: [merklee@missouri.edu](mailto:merklee@missouri.edu)  
 URL: <http://faculty.missouri.edu/~merklee/>

Column Name	Row Name	lme4	lavaan	Abs(diff)
(Intercept)	(Intercept)	43.99	43.99	0.00
Days	(Intercept)	-1.37	-1.37	0.00
cov_Subject.(Intercept)	(Intercept)	0.00	0.00	0.00
cov_Subject.Days.(Intercept)	(Intercept)	0.00	0.00	0.00
cov_Subject.Days	(Intercept)	0.00	0.00	0.00
residual	(Intercept)	0.00	0.00	0.00
(Intercept)	Days	-1.37	-1.37	0.00
Days	Days	2.26	2.26	0.00
cov_Subject.(Intercept)	Days	0.00	0.00	0.00
cov_Subject.Days.(Intercept)	Days	0.00	0.00	0.00
cov_Subject.Days	Days	0.00	0.00	0.00
residual	Days	0.00	0.00	0.00
(Intercept)	cov_Subject.(Intercept)	0.00	0.00	0.00
Days	cov_Subject.(Intercept)	0.00	0.00	0.00
cov_Subject.(Intercept)	cov_Subject.(Intercept)	70366.08	70366.15	0.07
cov_Subject.Days.(Intercept)	cov_Subject.(Intercept)	-2282.47	-2282.46	0.01
cov_Subject.Days	cov_Subject.(Intercept)	92.56	92.56	0.00
residual	cov_Subject.(Intercept)	-2058.08	-2058.08	0.00
(Intercept)	cov_Subject.Days.(Intercept)	0.00	0.00	0.00
Days	cov_Subject.Days.(Intercept)	0.00	0.00	0.00
cov_Subject.(Intercept)	cov_Subject.Days.(Intercept)	-2282.47	-2282.46	0.01
cov_Subject.Days.(Intercept)	cov_Subject.Days.(Intercept)	1838.33	1838.33	0.00
cov_Subject.Days	cov_Subject.Days.(Intercept)	-115.28	-115.28	0.00
residual	cov_Subject.Days.(Intercept)	324.96	324.96	0.00
(Intercept)	cov_Subject.Days	0.00	0.00	0.00
Days	cov_Subject.Days	0.00	0.00	0.00
cov_Subject.(Intercept)	cov_Subject.Days	92.56	92.56	0.00
cov_Subject.Days.(Intercept)	cov_Subject.Days	-115.28	-115.28	0.00
cov_Subject.Days	cov_Subject.Days	184.21	184.21	0.00
residual	cov_Subject.Days	-72.21	-72.21	0.00
(Intercept)	residual	0.00	0.00	0.00
Days	residual	0.00	0.00	0.00
cov_Subject.(Intercept)	residual	-2058.08	-2058.08	0.00
cov_Subject.Days.(Intercept)	residual	324.96	324.96	0.00
cov_Subject.Days	residual	-72.21	-72.21	0.00
residual	residual	5957.61	5957.61	0.00

Table 1: Comparison between `vcov.full.lmerMod()` output and **lavaan** `vcov()` output for the **SleepStudy** model. The first two columns describe the specific matrix entry being compared, the third and fourth columns show the estimates, and the fifth column shows the absolute difference.

Column Name	Row Name	lme4	lavaan	Abs(diff)
(Intercept)	(Intercept)	43.99	43.99	0.00
Days	(Intercept)	-1.37	-1.37	0.00
cov_Subject.(Intercept)	(Intercept)	-523.40	-523.41	0.01
cov_Subject.Days.(Intercept)	(Intercept)	-20.77	-20.77	0.00
cov_Subject.Days	(Intercept)	-5.92	-5.92	0.00
residual	(Intercept)	149.15	149.15	0.00
(Intercept)	Days	-1.37	-1.37	0.00
Days	Days	2.26	2.26	0.00
cov_Subject.(Intercept)	Days	-56.09	-56.09	0.00
cov_Subject.Days.(Intercept)	Days	0.18	0.18	0.00
cov_Subject.Days	Days	-1.98	-1.98	0.00
residual	Days	78.71	78.71	0.00
(Intercept)	cov_Subject.(Intercept)	-523.40	-523.41	0.01
Days	cov_Subject.(Intercept)	-56.09	-56.09	0.00
cov_Subject.(Intercept)	cov_Subject.(Intercept)	45232.13	45232.18	0.05
cov_Subject.Days.(Intercept)	cov_Subject.(Intercept)	1055.38	1055.38	0.00
cov_Subject.Days	cov_Subject.(Intercept)	427.39	427.39	0.00
residual	cov_Subject.(Intercept)	-27398.62	-27398.62	0.00
(Intercept)	cov_Subject.Days.(Intercept)	-20.77	-20.77	0.00
Days	cov_Subject.Days.(Intercept)	0.18	0.18	0.00
cov_Subject.(Intercept)	cov_Subject.Days.(Intercept)	1055.38	1055.38	0.00
cov_Subject.Days.(Intercept)	cov_Subject.Days.(Intercept)	1862.99	1862.99	0.00
cov_Subject.Days	cov_Subject.Days.(Intercept)	-89.28	-89.28	0.00
residual	cov_Subject.Days.(Intercept)	1214.37	1214.37	0.00
(Intercept)	cov_Subject.Days	-5.92	-5.92	0.00
Days	cov_Subject.Days	-1.98	-1.98	0.00
cov_Subject.(Intercept)	cov_Subject.Days	427.39	427.39	0.00
cov_Subject.Days.(Intercept)	cov_Subject.Days	-89.28	-89.28	0.00
cov_Subject.Days	cov_Subject.Days	137.89	137.89	0.00
residual	cov_Subject.Days	-492.56	-492.56	0.00
(Intercept)	residual	149.15	149.15	0.00
Days	residual	78.71	78.71	0.00
cov_Subject.(Intercept)	residual	-27398.62	-27398.62	0.00
cov_Subject.Days.(Intercept)	residual	1214.37	1214.37	0.00
cov_Subject.Days	residual	-492.56	-492.56	0.00
residual	residual	43229.03	43229.03	0.00

Table 2: Comparison of the **SleepStudy** sandwich estimator obtained from our **lmerMod** code with the analogous estimator obtained from **lavaan**. The first two columns describe the specific matrix entry being compared, the third and fourth columns show the estimates, and the fifth column shows the absolute difference.